This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

Claim 1 (Currently Amended): A method of validating and dispatching an event, comprising:

 generating a list of valid exception handlers, said list <u>of valid exception handlers located in a</u>

 protected <u>area</u> ~~from alteration~~ during program execution;

 receiving an event;

 determining an exception handler for the event;

 determining if the exception handler is valid by comparing the exception handler to said list

 of valid exception handlers and determining if the exception handler is unaltered; and

 otherwise determining that the exception handler is invalid; and

 executing the exception handler if the exception handler is valid.

Claim 2 (Canceled)

Claim 3 (Previously Presented): The method of claim 1, further comprising one of receiving and generating the list of valid exception handlers.

Claim 4 (Original): The method of claim 1, further comprising retrieving a list of valid exception handlers from a storage device and comparing the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

Claim 5 (Original): The method of claim 1, further comprising generating a list of valid exception handlers by compiling code into at least one of an object file and an image.

Claim 6 (Original): The method of claim 1, further comprising compiling code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

Claim 7 (Original): The method of claim 1, further comprising, if the exception handler is valid, determining whether the exception handler handles the event, and if so, executing the exception handler, and otherwise, retrieving a second exception handler from information on a stack and continuing processing with determining if the second exception handler is valid.

Claim 8 (Previously Presented): The method of claim 1, further comprising terminating the method if the exception handler is invalid.

Claim 9 (Previously Presented): The method of claim 1, further comprising generating an error message if the exception handler is invalid.

Claim 10 (Original): The method of claim 1, further comprising, if the exception handler is valid, verifying other data for the event.

Claim 11 (Original): The method of claim 10, wherein the other data comprises pointer data.

Claim 12 (Currently Amended): A computer-readable storage medium having stored thereon computer-executable instructions for performing a method of validating and dispatching an event, the method comprising:

    generating a list of valid exception handlers, said list <u>of valid exception handlers located in a</u>
        protected <u>area</u> ~~from alteration~~ during program execution;
    receiving an event;
    determining an exception handler for the event;
    determining if the exception handler is valid by comparing the exception handler to said list
        of valid exception handlers and determining if the exception handler is unaltered; and
        otherwise determining that the exception handler is invalid; and
    executing the exception handler if the exception handler is valid.

Claim 13 (Canceled)

Claim 14 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for one of receiving and generating the list of valid exception handlers.

Claim 15 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for retrieving a list of valid exception handlers from a storage device and comparing the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

Claim 16 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for generating a list of valid exception handlers by compiling code into at least one of an object file and an image.

Claim 17 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for compiling code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

Claim 18 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for, if the exception handler is valid, determining whether the exception handler handles the event, and if so, executing the exception handler, and otherwise, retrieving a second exception handler from information on a stack and continuing processing with determining if the second exception handler is valid.

Claim 19 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for terminating the method if the exception handler is invalid.

Claim 20 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for generating an error message if the exception handler is invalid.

Claim 21 (Previously Presented): The computer-readable storage medium of claim 12, having further computer-executable instructions for, if the exception handler is valid, verifying other data for the event.

Claim 22 (Previously Presented): The computer-readable storage medium of claim 21, wherein the other data comprises pointer data.

Claim 23 (Currently Amended): A system for validating an event to be dispatched, comprising:
    a processor that receives an event; and
    an exception dispatcher system that generates a list of valid exception handlers, said list of
        valid exception handlers located in a protected area from alteration during program
        execution, determines an exception handler for the event, determines if the exception
        handler is valid by comparing the exception handler to said list of valid exception
        handlers and determining if the exception handler is unaltered, and otherwise determining
        that the exception handler is invalid; and executes the exception handler if the exception
        handler is valid.

Claim 24 (Canceled)

Claim 25 (Original): The system of claim 23, further comprising a storage device that stores a list of valid exception handlers, and the exception dispatcher system retrieves the list of valid exception handlers from the storage device and compares the exception handler to the list of valid exception handlers in determining if the exception handler is valid.

Claim 26 (Original): The system of claim 23, further comprising at least one of a compiler and an assembler that generates a list of valid exception handlers in at least one of an object file and an image.

Claim 27 (Original): The system of claim 26, further comprising a storage device that stores the list of valid exception handlers.

Claim 28 (Original): The system of claim 23, further comprising a compiler and a linker that compiles code to produce an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

Claim 29 (Original): The system of claim 23, wherein the exception dispatcher system, if the exception handler is valid, determines whether the exception handler handles the event, and if so, executes the exception handler, and otherwise, retrieves a second exception handler from information on a stack and continues processing with determining if the second exception handler is valid.

Claim 30 (Previously Presented): The system of claim 23, wherein the exception dispatcher system terminates processing if the exception handler is invalid.

Claim 31 (Previously Presented): The system of claim 23, wherein the exception dispatcher system generates an error message if the exception handler is invalid.

Claim 32 (Original): The system of claim 23, further comprising a linker that creates an image based on at least one object file received from at least one of a compiler and an assembler, and provides the image to the exception dispatcher system.

Claim 33 (Original): The system of claim 32, wherein the linker produces an executable that is marked with an identifier indicating that the executable is safe with respect to a list of valid exception handlers.

Claim 34 (Original): The system of claim 23, wherein the exception dispatcher system, if the exception handler is valid, verifies other data for the event.

Claim 35 (Original): The system of claim 34, wherein the other data comprises pointer data.